Introduction to Big Data Assignment-6

Name: Faizan Mulla
Roll No: 21F1003885

Problem Statement

Count the number of lines in a file in real-time using Google Cloud Functions and Pub Sub. For this, you will need an input file on which you will count the lines.

- Write a Google Cloud Function (GCF) that gets triggered whenever a file is added to a bucket. It should publish that file name to a topic in Pub Sub.

- Write a Python program that subscribes to that Pub Sub topic, picks up the file name, reads the file, counts the lines and prints it out.

Trigger the GCF by placing the input file in the bucket.

Solution

Environment Setup

1. GCP Setup
   - Set up a new project with default service account configurations
   - Enabled necessary APIs (Compute Engine and Dataproc)

2. Cloud Storage Setup
   - Created bucket "**ibd-ga6-bucket**"
   - **Region**: asia-south1
   - Configured with standard storage class

3. Pub/Sub Topic & Subscription
   - Navigate to the Analytics section & then to Pub/Sub section or search directly for Pub/Sub in the main search bar.
   - Go to "**Topics**" and create a Topic named "**ibd-ga6-topic**" (rest all settings are default)

- Go to "**Subscriptions**" and create a Subscription named "**ibd-ga6-subscription**"
    - Choose the topic, which you just created i.e. **"ibd-ga6-topic**"
    - Ensure the delivery type is **'Pull'**
    - Rest all settings remain default.

Implementation Details

1. **Write a Google Cloud Function (GCF)**

   Create a Cloud Function that triggers on file upload to the **"iitm-ibd-ga6"** storage bucket and then publishes the file name to the **"ibd-ga6-topic"** pub/sub topic.

   Settings:

   - Environment: Cloud Run Function
   - Function Name: **ibd-ga6-function**
   - Region: **asia-south-1**
   - Trigger Settings:
     - Trigger Type: Cloud Run Storage
     - Event Type: google.cloud.storage.object.v1.finalized
     - Bucket: Choose the one you just created. In my case it is → **iitm-ibd-ga6**
   - Optional Settings:
     - Runtime :
       - Memory Allocated : 256 MiB to 512 MiB
       - CPU: 0.167 to 0.583

   - Click on Next
   - Now, set the runtime environment to **Python 3.9** and define the entry point as **`process_file`**.

- Add code to the main.py

```python
import functions_framework
import json
from google.cloud import pubsub_v1


project_id = "intro-to-big-data-439410"
topic_id = "ibd-ga6-topic"

# Initialize Pub/Sub client
publisher = pubsub_v1.PublisherClient()
topic_path = publisher.topic_path(project_id, topic_id)


# Triggered by a change in a storage bucket
Tabnine | Edit | Test | Explain | Document | Ask
@functions_framework.cloud_event
def process_file(cloud_event):
    data = cloud_event.data

    event_id = cloud_event["id"]
    event_type = cloud_event["type"]

    bucket = data["bucket"]
    name = data["name"]
    metageneration = data["metageneration"]
    timeCreated = data["timeCreated"]
    updated = data["updated"]

    print(f"Event ID: {event_id}")
    print(f"Event type: {event_type}")
    print(f"Bucket: {bucket}")
    print(f"File: {name}")
    print(f"Metageneration: {metageneration}")
    print(f"Created: {timeCreated}")
    print(f"Updated: {updated}")

    publish_to_pubsub(bucket, name)


# Function to publish file name to Pub/Sub topic
Tabnine | Edit | Test | Explain | Document | Ask
def publish_to_pubsub(bucket_name, file_name):
    message_data = file_name.encode("utf-8")
    future = publisher.publish(topic_path, data=message_data)
    print(f"Published message {future.result()} for file {file_name} to Pub/Sub topic.")
```

- And, now create a requirements.txt file and add the following lines to it:

  *functions-framework==3.**
  google-cloud-storage
  google-cloud-pubsub
  google-api-core
  google-cloud-functions
  google-auth

- Finally deploy the function

==========

2. **Write the Python program to subscribe to the Pub/Sub topic**
   Create a Python script that subscribes to the ibd-ga6-subscription, then reads the file from the bucket, counts lines and then prints the result.

   For this create a Virtual Machine Instance.

   Step 1: VM instance configuration:
   - Navigate to Compute Engine and then to VM instances.
   - Main settings:
     - Name: **ibd-ga6-vm**
     - Region: **asia-south-1**
     - Zone: **asia-south-1-a**
   - Keep Machine Configuration settings as default (or you can change it based on your requirement)
   - Identity and Access Scopes settings (**IMP**)
     - Choose this: Allow full access to all Cloud APIs
   - Firewall Settings
     - Check these boxes: Allow HTTP traffic & Allow HTTPS traffic

   - Click on “Create”

Step 2: ***Environment Preparation / VM setup***

Click the "SSH" button next to your VM. This opens a browser-based terminal

Google Cloud SDK Configuration and authenticate:

- Now, in the SSH terminal, run these commands:

  ```
  curl -O
  https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloudsdk-xxx
  -linux-x86_64.tar.gz

  tar -xf google-cloud-sdk-xxx-linux-x86_64.tar.gz

  ./google-cloud-sdk/install.sh
  ./google-cloud-sdk/bin/gcloud init
  ./google-cloud-sdk/bin/gcloud auth application-default login
  ```

- Type "Y" to log in
- Click on the link it shows
- Log in with your Google account
- Copy the verification code shown
- Paste it back in the terminal
- Select your project number when asked
- Choose your default region. For me it is: "**asia-south-1-a**"

Create Virtual Environment

- First install required packages:

  sudo apt-get update

  sudo apt-get install python3-venv python3-pip

- Create a directory for your project:
  mkdir ga6
  cd ga6

- Create a virtual environment
  python3 -m venv venv

- Activate the virtual environment:
  source venv/bin/activate

- Now install the Google Cloud Storage package:
  pip3 install google-cloud-storage

- **NOTE**: Every time you log into your VM and want to run the script, you'll need to:
  cd ga1
  source venv/bin/activate

  Now, you have to upload 2 files, "*subscription.py*" and "*requirements.txt*"

- **Method 1:**
  - In the top right corner, click on "Upload File" and choose the Python script 'subscription.py' and 'requirements.txt' from your computer. These files will be in your home directory now.
  - Then, move them to the ga6 folder using: "*mv subscription.py requirements.txt ga6/*" command.

- **Method 2:**
  - Using the editor itself to create files and enter code.

Now run the command '*pip install -r requirements.txt'* to install the dependencies listed in the requirements.txt file.

Finally, we have to run the Python file using the commands '*python3 subscription.py*'

```python
from google.cloud import pubsub_v1, storage
import time


Tabnine | Edit | Test | Explain | Document | Ask
def callback(message):
    """Callback function to handle incoming messages."""
    filename = message.data.decode("utf-8")
    print(f"Received message for file: {filename}")
    try:
        # Adding a delay to ensure the file is available in storage
        time.sleep(5)
        lines = lines_counter(filename)
        print(f"The number of lines in {filename} are {lines}")
    except Exception as e:
        print(f"Error processing file {filename}: {e}")
    message.ack()


Tabnine | Edit | Test | Explain | Document | Ask
def lines_counter(filename):
    """Count the number of lines in a file stored in Google Cloud Storage."""
    client = storage.Client()
    bucket = client.get_bucket("iitm-ibd-ga6")
    blob = bucket.blob(filename)
    with blob.open("r") as file:
        lines = len(file.readlines())
    return lines


Tabnine | Edit | Test | Explain | Document | Ask
def subscribe_pub_sub(project_id, subscription_name):
    """Subscribe to a Pub/Sub subscription and listen for messages."""
    subscriber = pubsub_v1.SubscriberClient()
    subscription_path = subscriber.subscription_path(project_id, subscription_name)
    print(f"Subscribing to {subscription_path}...")

    # Listen for messages with an attached callback
    streaming_pull_future = subscriber.subscribe(subscription_path, callback=callback)
    print("Listening for messages... Press Ctrl+C to exit.")

    try:
        streaming_pull_future.result()
    except KeyboardInterrupt:
        print("Subscriber script terminated by user.")
        streaming_pull_future.cancel()
    except Exception as e:
        print(f"Error while listening for messages: {e}")
        streaming_pull_future.cancel()


if __name__ == "__main__":
    project_id = "intro-to-big-data-439410"
    subscription_name = "ibd-ga6-subscription"
    subscribe_pub_sub(project_id, subscription_name)
```

Execution Process

Now, you just have to trigger the cloud function by uploading a text file (whose line count you want) to the GCS bucket (iitm-ibd-ga6)

Steps:

- Go to Storage Bucket and click on “Upload files”.
- Now select the input text file.\

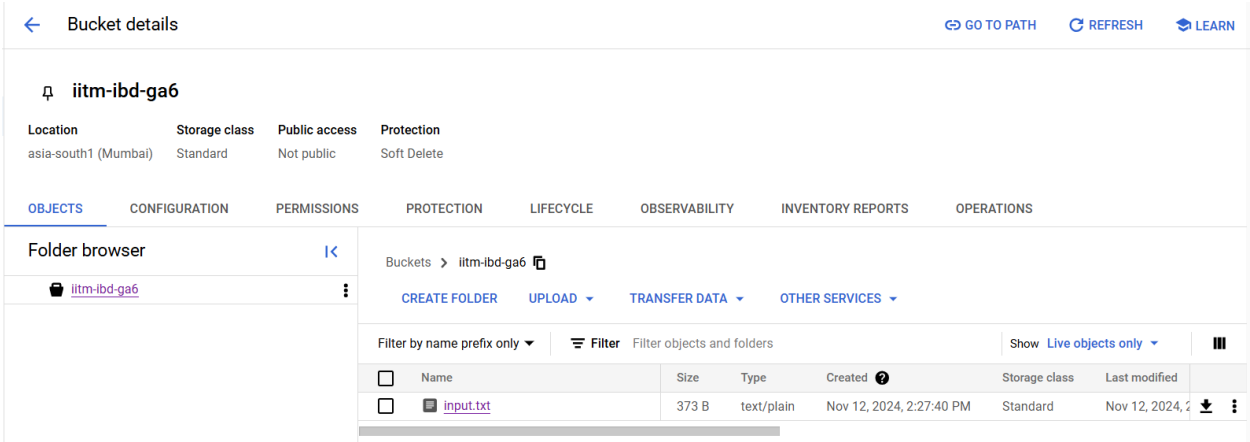
Come back to the SSH terminal to see the required result.

Results

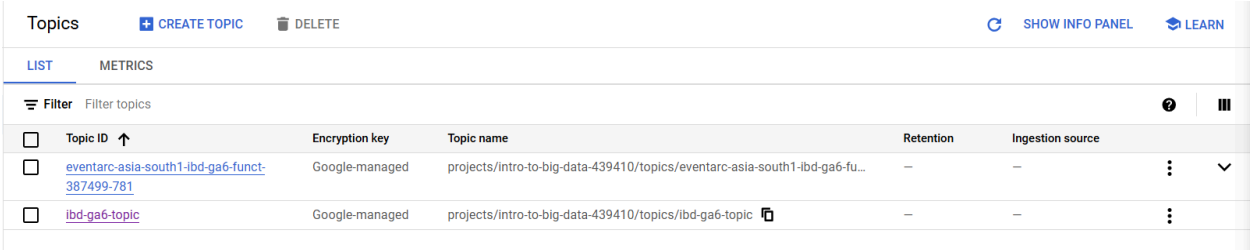
```
faizanamulla69@ibd-ga6-vm:~$ cd ga6
faizanamulla69@ibd-ga6-vm:~/ga6$ source venv/bin/activate
(venv) faizanamulla69@ibd-ga6-vm:~/ga6$ ls
requirements.txt  subscription.py  venv
(venv) faizanamulla69@ibd-ga6-vm:~/ga6$ python3 subscription.py
Subscribing to projects/intro-to-big-data-439410/subscriptions/ibd-ga6-subscription...
Listening for messages... Press Ctrl+C to exit.
Received message for file: input.txt
The number of lines in input.txt are 9
```

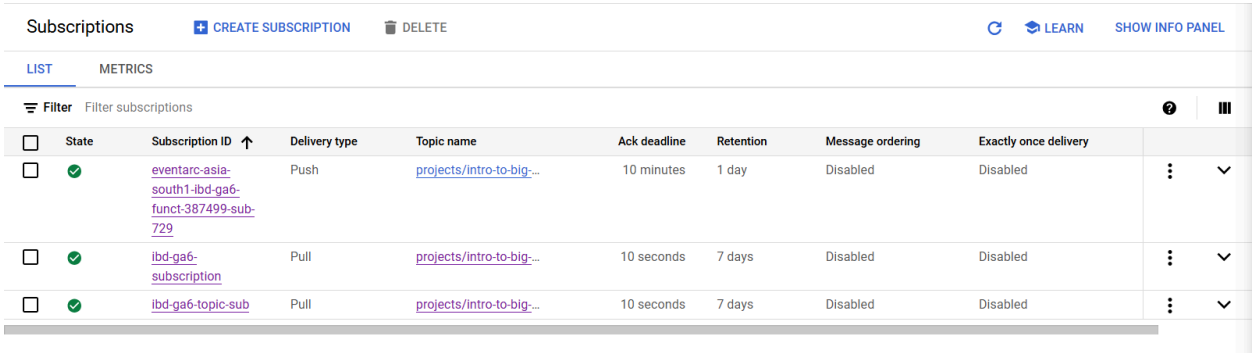
Relevant Screenshots

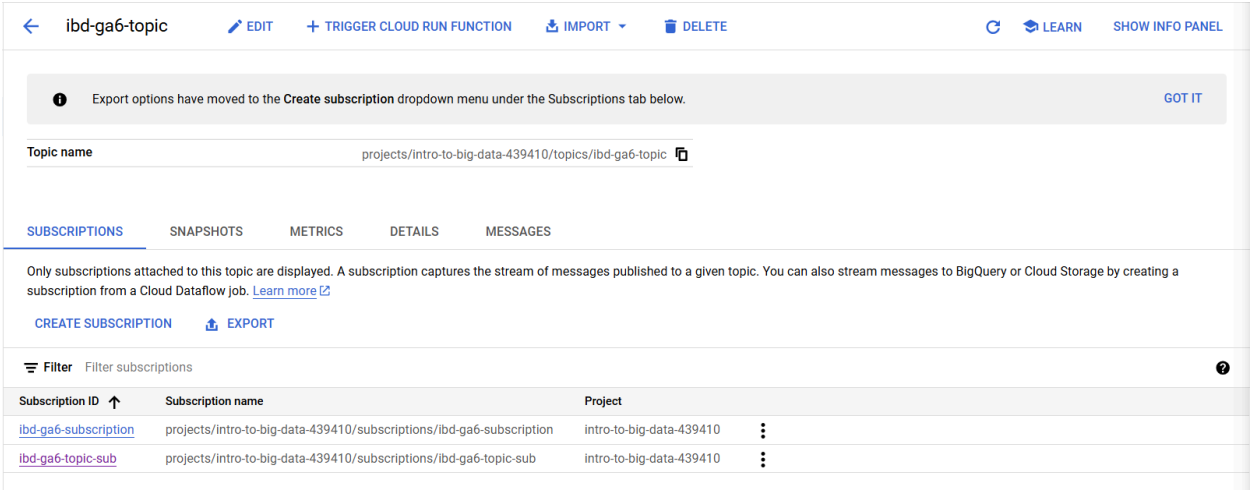
1. Cloud Storage Bucket Contents



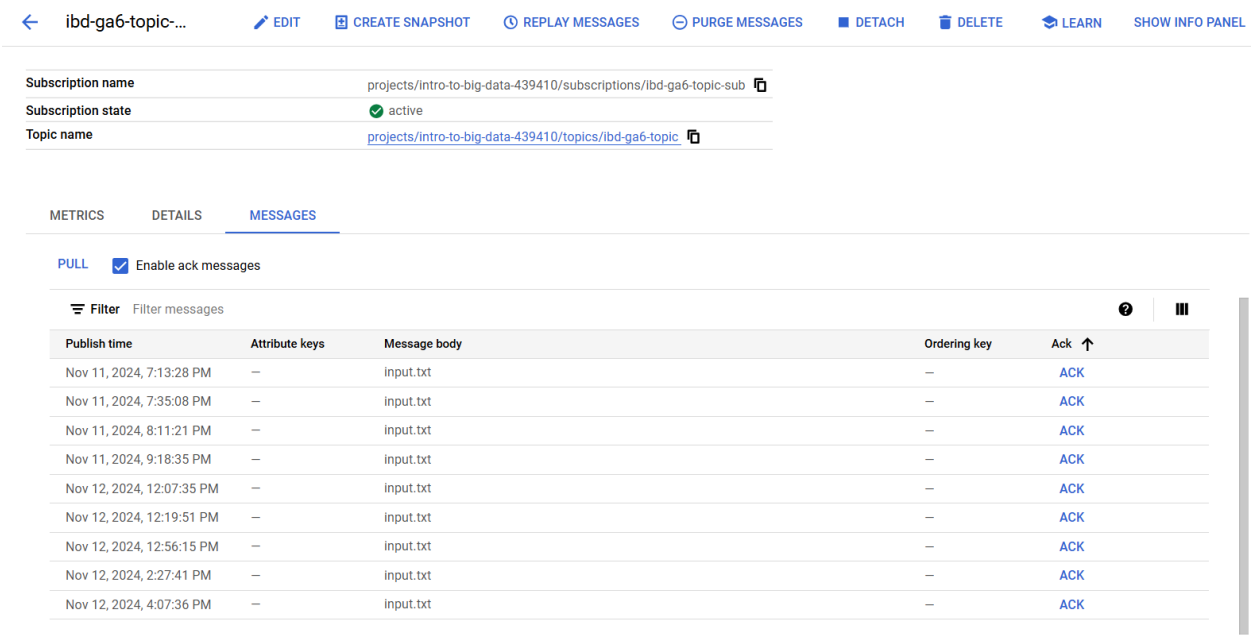
2. Pub/Sub Topic & Subscription






ibd-ga6-topic
EDIT
TRIGGER CLOUD RUN FUNCTION
IMPORT
DELETE
LEARN
SHOW INFO PANEL
Export options have moved to the Create subscription dropdown menu under the Subscriptions tab below.
GOT IT
Topic name
projects/intro-to-big-data-439410/topics/ibd-ga6-topic
SUBSCRIPTIONS
SNAPSHOTS
METRICS
DETAILS
MESSAGES
Only subscriptions attached to this topic are displayed. A subscription captures the stream of messages published to a given topic. You can also stream messages to BigQuery or Cloud Storage by creating a subscription from a Cloud Dataflow job. Learn more
CREATE SUBSCRIPTION
EXPORT
Filter
Filter subscriptions
Subscription ID
Subscription name
Project
ibd-ga6-subscription
projects/intro-to-big-data-439410/subscriptions/ibd-ga6-subscription
intro-to-big-data-439410
ibd-ga6-topic-sub
projects/intro-to-big-data-439410/subscriptions/ibd-ga6-topic-sub
intro-to-big-data-439410

3. Messages


ibd-ga6-topic-...
EDIT
CREATE SNAPSHOT
REPLAY MESSAGES
PURGE MESSAGES
DETACH
DELETE
LEARN
SHOW INFO PANEL
Subscription name
projects/intro-to-big-data-439410/subscriptions/ibd-ga6-topic-sub
Subscription state
active
Topic name
projects/intro-to-big-data-439410/topics/ibd-ga6-topic
METRICS
DETAILS
MESSAGES
PULL
Enable ack messages
Filter
Filter messages
Publish time
Attribute keys
Message body
Ordering key
Ack
Nov 11, 2024, 7:13:28 PM
Nov 11, 2024, 7:35:08 PM
Nov 11, 2024, 8:11:21 PM
Nov 11, 2024, 9:18:35 PM
Nov 12, 2024, 12:07:35 PM
Nov 12, 2024, 12:19:51 PM
Nov 12, 2024, 12:56:15 PM
Nov 12, 2024, 2:27:41 PM
Nov 12, 2024, 4:07:36 PM
input.txt
ACK

4. **GCF function + Trigger Settings**